

Gabriel André Melo de Oliveira Silva
Laís Carraro Leme Cavaleiro
Matheus Morandino di Giovanni Carneiro

Composição algorítmica de música popular com Modelos Ocultos de Markov

Brasil

2019

Gabriel André Melo de Oliveira Silva
Laís Carraro Leme Cavaleiro
Matheus Morandino di Giovanni Carneiro

Composição algorítmica de música popular com Modelos Ocultos de Markov

Universidade de São Paulo
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES

Brasil
2019

Resumo

O seguinte trabalho teve como objetivo a criação de um modelo computacional capaz de gerar composições musicais em diferentes gêneros da música popular. O modelo escolhido foi o Modelo Oculto de Markov (HMM), um modelo generativo equivalente a um autômato probabilístico. O modelo foi alimentado com diferentes conjuntos de dados: um conjunto Geral (com todas as músicas utilizadas), três conjuntos de diferentes gêneros musicais (anime, musical e pop), e dois conjuntos de artistas específicos (ABBA e Frozen). Os resultados obtidos não apresentaram nenhuma tipo de coerência global, já que os HMMs não possuem memória de estados anteriores. Pôde-se observar, no entanto, uma grande quantidade de trechos coerentes, os quais apresentaram algumas peculiaridades de cada gênero musical. Todo o conteúdo do trabalho encontra-se neste [repositório do Github](#).

Palavras-chave: Música, Cifras, HMM, Composição, Algorítmica.

Sumário

1	Introdução	4
1.1	Motivação para a escolha do tema	4
1.2	Motivação para a escolha do modelo	4
2	Fundamentação Teórica	5
2.1	Hidden Markov Models	5
3	Metodologia	6
3.1	Conjuntos de dados	6
3.2	Implementação do modelo	7
3.3	Decodificação dos resultados	8
4	Resultados	10
4.1	Geral	10
4.2	Anime	10
4.3	Musical	11
4.4	Pop	12
4.5	Frozen	12
4.6	ABBA	12
5	Discussão	14
	Conclusão	15
	Referências	16

1 Introdução

1.1 Motivação para a escolha do tema

A área de Recuperação de Informação em Música, ou Music Information Retrieval (MIR), tem grande interesse no estudo da automação da composição musical. Grande parte dos estudos existentes, no entanto, têm como foco a harmonia clássica para a composição de música de concerto, também chamada de música erudita e, popularmente, música clássica.

Dessa escassez de estudos sobre o tema, surgiu o interesse deste trabalho de estudar a automação da composição de música popular (mais especificamente, de cifras), utilizando a harmonia funcional ao invés da tradicional.

1.2 Motivação para a escolha do modelo

Os Modelos Ocultos de Markov, ou Hidden Markov Models (HMM), são um dos modelos generativos mais comuns. De maneira simplificada, são análogos a autômatos probabilísticos (DUPONT; DENIS; ESPOSITO, 2005). Uma das aplicações comuns dos HMMs são no campo de geração de textos. Dado um estado inicial, ou seja, uma palavra ou um conjunto inicial de palavras, esse modelo é capaz de prever a próxima palavra através das transições geradas pelo conjunto de treinamento. O resultado é um texto que se assemelha aos textos do conjunto de treinamento, mas não é igual a nenhum deles.

Diversas técnicas de Processamento de Linguagem Natural (PLN) têm aplicação na área de MIR, já que a notação de acordes se assemelha a palavras, e uma sequência de acordes pode ser escrita como diversas "palavras" separadas por espaço, assim como nos textos. Além disso, a transição entre acordes deve fazer sentido, assim como a transição entre palavras. Essas transições entre dois (ou mais) acordes são chamadas de progressões (MED, 1996).

Desse modo, a principal motivação para a escolha do modelo foi gerar sequências de acordes com progressões coerentes e características de diferentes gêneros da música popular. Como os HMMs não possuem memória a longo prazo, no entanto, não se espera que haja qualquer tipo de coerência global no resultado gerado.

2 Fundamentação Teórica

2.1 Hidden Markov Models

Um modelo de Markov oculto (HMM) é um modelo estatístico de Markov no qual um sistema que está sendo modelado é considerado um processo de Markov com estados não observados (ocultos) (RABINER, 1989). Os modelos de Markov ocultos são especialmente conhecidos por sua aplicação no reconhecimento de padrões temporais, como fala, escrita à mão, reconhecimento de gestos (STARNER, 1995), partitura musical (PARDO; BIRMINGHAM, 2005), entre outros.

Nos modelos de Markov mais simples (como uma cadeia de Markov (BAUM L. E.; PETRIE, 2011)), o estado é diretamente visível ao observador e, portanto, as probabilidades de transição de estado são os únicos parâmetros. Em um modelo oculto de Markov (RABINER, 1989), o estado não é diretamente visível, mas a saída, dependente do estado, é visível. Cada estado tem uma distribuição de probabilidade entre os possíveis tokens de saída e, portanto, a sequência de tokens gerados por um HMM fornece algumas informações sobre a sequência de estados. O adjetivo "oculto" refere-se à sequência de estados pela qual o modelo passa, não aos parâmetros do modelo: o modelo ainda é referido como um modelo de Markov "oculto", mesmo que esses parâmetros sejam conhecidos exatamente. No geral, um dado HMM possui três problemas básicos (RABINER, 1989) relacionados na estimação de seus parâmetros e sua usabilidade, sendo eles:

1. Dados os dados de observação O e os parâmetros do modelo λ , podemos calcular as probabilidades das observações $P(O | \lambda)$?
2. Dados os dados de observação O e os parâmetros do modelo λ , podemos encontrar o melhor estado oculto sequência de O ?
3. Dada a observação O , podemos encontrar os parâmetros do modelo λ ?

Por mais que existam formas mais formais para estimar os parâmetros de λ , como o algoritmo de Baum–Welch (RABINER, 1989), para o problema 3, neste trabalho foi tomada uma abordagem diferente para definir os parâmetros. Detalhes sobre a específica implementação serão dados no capítulo de Metodologia.

3 Metodologia

3.1 Conjuntos de dados

A falta de um conjunto de dados tão específico e com dados no formato ideal para o estudo levou à necessidade da criação de um novo dataset. As cifras foram retiradas de maneira automatizada de páginas do site [Cifra Club](#), através da utilização de *expressões regulares*. A escolha e o download das páginas foram feitos, no entanto, de forma manual. Por esse motivo, os diferentes conjuntos de dados possuem tamanho reduzido, o que não afeta, todavia, a performance do modelo.

Uma das vantagens de sua escolha é a de que ela é capaz de gerar resultados significativos usando um conjunto de dados relativamente pequeno, se comparado com o que é necessário para modelos mais famosos de geração de texto, como um Long Short Term Memory (LSTM) (PANZNER; CIMIANO, 2016). Contudo, como a seleção de parâmetros apenas estuda a distribuição de tokens no dataset, um aumento no conjunto de treino tende a não causar uma significativa melhora nos resultados.

O uso de expressões regulares se deu na implementação de um algoritmo capaz de selecionar apenas as cifras da página em `.html`, ao encontrar todas as ocorrências de uma *tag* específica em que os acordes são anotados. Tal algoritmo foi implementado com a biblioteca `re` da linguagem Python, e encontra-se em [scripts/dataseter.py](#). O processamento do arquivo `.html` resulta num arquivo `.txt` com apenas a sequência de acordes da música, separados por espaço.

Um cuidado extra que se teve na escolha das cifras para os conjuntos de dados foi o de deixar todas elas na mesma tonalidade. Desse modo, as transições entre os acordes estarão sempre numa mesma escala, e poderemos analisá-las como se fossem análogas a transições entre graus. Caso houvesse uma música em dó maior e outra em sol maior, por exemplo, na primeira música a sequência "G C" seria uma cadência perfeita V-I, enquanto na segunda música, a mesma sequência seria uma progressão I-IV. (MED, 1996)

Assim, garantimos uma "normalização" dos dados, e podemos nos certificar de que uma mesma sequência de acordes será interpretada como uma única progressão. Escolheu-se arbitrariamente a tonalidade de dó maior (C, na notação de cifra).

Foram criados, no total, 6 conjuntos de dados, distribuídos em 3 categorias:

1. **Geral:** a união de todos os conjuntos citados abaixo (disponível em [datasets/tudo](#)).
2. **Gêneros musicais:** temos 3 conjuntos nessa categoria, especificados a seguir.

- a) **Anime:** um dataset que consta apenas de músicas provenientes de animes (disponível em [datasets/anime](#)).
 - b) **Musicals:** este conjunto, por sua vez, é composto por músicas de filmes musicais, principalmente da *Disney* (disponível em [datasets/musicals](#)).
 - c) **Pop:** por fim, este dataset é composto por músicas do gênero Pop (disponível em [datasets/pop](#)).
3. **Artista específico:** nesta última categoria, filtramos apenas um artista dentro das categorias de gêneros musicais, como listados a seguir.
- a) **ABBA:** dentro do gênero Pop, selecionamos apenas as músicas de ABBA (disponível em [datasets/pop/abba](#)).
 - b) **Frozen:** agora dentro da pasta Musicals, filtramos só as músicas do filme *Frozen* (disponível em [datasets/musicals/frozen](#)).

Como cada um dos gêneros musicais apresentam estruturas harmônicas muito diferentes entre si, espera-se que o modelo treinado com o dataset **Geral** terá o pior desempenho. No entanto, como esse conjunto será o maior de todos, o teste ainda é válido.

Como já comentado, há uma forte correlação harmônica dentre as músicas que se encontram num mesmo gênero musical, mesmo que tal correlação não seja tão forte quanto as músicas de um mesmo artista. Os datasets mais específicos, no entanto, são também os menores. Por isso, espera-se que o modelo treinado com os datasets de artistas específicos tenham um desempenho mediano, e que o treinado com o dos gêneros musicais terá a melhor performance.

Entre os gêneros musicais citados, é válido apontar que o gênero **Pop** é o mais homogêneo e "monótono" harmonicamente, já que, historicamente, apresenta fórmulas repetitivas e comuns para serem facilmente digeridos e atrair as massas (NUNESA J. C.; ORDANINIB; VALSESIA, 2015).

3.2 Implementação do modelo

Inicialmente o modelo percorre os arquivos de treino, identificando os tokens (neste caso, os acordes) e usando o caracter de espaço como marcação. Cada leitura de arquivo é adicionada no dicionário de acordes presentes no dataset. Com isso em mente, é criada uma distribuição baseada na frequência de transição entre os tokens, com cada transição dependente do token anterior. Tal processo é repetido para cada arquivo do conjunto,

criando, assim, uma sequência de distribuições para cada um. Sabendo disso, cada distribuição é ordenada em função da similaridade das suas distribuições, com distribuições que possuem mais acordes em comum recebendo uma maior chance de transição.

Após a geração do modelo, a parte generativa é iniciada. O usuário deve fornecer ao modelo uma sequência inicial de acordes (uma "seed"), que definirá o estado inicial do modelo. Para fins de comparação, a seed escolhida para a geração de resultados em diferentes conjuntos de dados foi a mesma. A sequência escolhida foi "C Am F G", que resulta em uma transição "I-vi-IV-V", comum na música popular (MED, 1996).

A partir daí, o HMM seleciona um acorde de acordo com a distribuição formada, e assim por diante, até que um número pré-definido de observações seja gerado. Para este trabalho, o modelo foi ajustado para retornar 60 observações, ou seja, 60 acordes, que resultará em músicas de por volta de 1 minuto.

Tal forma de implementação é significativamente mais simples do que a maioria dos modelos usados em outras pesquisas. O algoritmo não é tão bem sucedido em capturar os detalhes de progressão de acordes, ou ser capaz de se manter em uma linha melódica, o que é ainda mais agravado devido à propriedade de falta de memória inerente aos HMMs. Por outro lado, uma vantagem da implementação usada para seleção dos parâmetros, como já comentado na seção anterior, é que ela é capaz de gerar resultados minimamente coerentes com um conjunto de dados relativamente pequeno (PANZNER; CIMIANO, 2016).

A implementação do modelo pode ser encontrada em [scripts/Model.py](#).

3.3 Decodificação dos resultados

Como o modelo é alimentado por arquivos `.txt`, ele retorna `strings` de acordes, ou seja, basicamente, o output do modelo é apenas texto. No entanto, como o objetivo do trabalho é a composição algorítmica de música, existe a necessidade de se escutar a música gerada pelo modelo.

Por tal motivo, foi criado um decodificador dos resultados do modelo para arquivos no formato `MusicXML`, com a extensão `.xml`. O formato em questão foi desenvolvido especificamente para a estruturação de dados musicais, e pode ser interpretado por softwares de notação musical. O software utilizado foi o [MuseScore](#), e lá os arquivos gerados foram exportados para o formato `.pdf` (partitura) e `.wav` (áudio).b

O decodificador foi feito com o auxílio da biblioteca [music21](#) de Python, desenvolvida pelo MIT para facilitar a criação e manipulação de arquivos `MusicXML`. O decodificador é capaz de transformar cifras (ou seja, o texto de notação de acordes) em acordes estruturados em `MusicXML`, e retornar um arquivo `.xml` contendo a música gerada pelo modelo. Ele consegue manipular acordes maiores, menores, diminutos, aumentados, com

baixo diferente da nota fundamental e acordes com sétima maior. Não é capaz, no entanto, de trabalhar com sétimas menores, nonas, décimas primeiras e outras variações de acordes.

É válido notar, também, que para fins de melhor entendimento da música e das transições entre os graus, o decodificador repete uma vez cada acorde. A implementação do decodificador pode ser encontrada em [scripts/chord_decoder.py](#).

4 Resultados

A análise dos resultados será feita através de uma análise harmônica simplificada de cada música gerada. Tal análise será baseada na Harmonia Funcional, como já especificada no texto. Os critérios analisados serão a formação de progressões coerentes, a repetição de estruturas harmônicas e a coerência local e global da música (MED, 1996). Recomenda-se que as avaliações sejam lidas observando a partitura de cada música, já que serão referenciados números de compassos específicos.

4.1 Geral

Nesta seção, será analisada a música formada pelo modelo treinado com o conjunto Geral, que se encontra em [datasets/tudo](#). A partitura resultante gerada pelo modelo pode ser encontrada em [arquivos_resultantes/partituras/tudo.pdf](#), e o áudio referente a ela pode ser encontrado em [arquivos_resultantes/audios/tudo.wav](#).

A primeira sequência de acordes coerente se encontra nos compassos 5 e 6 (I-vi-ii-III), que tem uma estrutura muito semelhante à da seed inserida (I-vi-IV-V). A tensão causada pelo grau III no fim do compasso 6, no entanto, não é resolvida de maneira coerente no compasso 7 (MED, 1996). Há uma repetição excessiva da sequência VI-V durante os compassos 8, 9 e 10 que, apesar de representar repetições comuns na música popular, causam monotonia.

A partir do compasso 11, o modelo foi capaz de gerar progressões coerentes até o compasso 14, mas ela é quebrada no início do 15. Os compassos 19 e 20 formam uma progressão comum na música popular, iii-IV-vi-V (MED, 1996).

Após isso, não há coerência explícita até o final da música que, inclusive, não termina com uma cadência perfeita, recurso usado na música para dar ao ouvinte a sensação de "conclusão". Por isso, a música parece inacabada, apesar de a última nota ser o grau I, ou seja, a nota fundamental da escala (MED, 1996). Tal fato, no entanto, pode ser uma mera coincidência, já que o HMM não trata para que a música gerada termine numa cadência, ele apenas para de gerar observações quando alcança o número especificado pelo usuário.

4.2 Anime

Analisaremos, agora, a música gerada a partir do dataset de músicas de Animes, que se encontra em [datasets/anime](#). A partitura gerada pode ser encontrada em

[arquivos_resultados/partituras/anime.pdf](#), e o áudio referente à partitura encontra-se em [arquivos_resultados/audios/anime.wav](#).

São encontradas progressões coerentes nos compassos 5 e 6, 9 e 10, 13 e 14. No compasso 15, encontramos o acorde B7, que representa o grau VII7, e causa uma tensão interessante que contrasta com o restante da música. Tal uso de acordes com sétima maior é muito comum em músicas deste gênero musical (JURKIEWICZB, 2019). Essa tensão, no entanto, não é resolvida de maneira coerente no compasso 16, o que causa no ouvinte uma quebra negativa de expectativa. Isso acontece novamente no compasso 28.

No compasso 21, a progressão vi-ii do compasso 20 é tensionada de maneira coerente, com iii-II e, apesar de ela não ser resolvida nos compassos subsequentes, há coerência nas progressões dos dois compassos seguintes, que seguem essa mesma estrutura (MED, 1996). Um ponto interessante é a criação de tensão no compasso 26, com o uso do grau iv7, e sua imediata resolução, com IV. Apesar de esta ser uma coerência local notável, não faz sentido comparado ao restante da música.

O compasso 30 traz a progressão IV-v-I-IV, que soa como uma cadência ao ouvinte, trazendo a sensação de conclusão (MED, 1996). Os compassos subsequentes, contudo, desfazem essa sensação.

4.3 Musical

A música analisada nesta seção será a gerada pelo modelo treinado com o dataset de músicas de musicais, presente em [datasets/musicals](#). A partitura pode ser encontrada em [arquivos_resultados/partituras/musical.pdf](#), e o áudio está localizado em [arquivos_resultados/audios/musical.wav](#).

Da metade do compasso 6 até a metade do compasso 8, a progressão formada soa ao ouvinte como um caminho muito natural que as notas seguem, até que a tensão do início do compasso 8 é quebrada pelo acorde de lá menor. Tais progressões são características de musicais, porque a harmonia tende a acompanhar o caminho criado pela melodia, nesse gênero musical, apresentando, inclusive, um maior número de inversões de acordes (FRANCESCHINA, 2015).

O modelo foi capaz de gerar uma estrutura chamada período, do compasso 13 ao 18, que é uma progressão de acordes que possui um início, um meio e um fim, como uma frase num texto. Para um ouvinte desavisado, a música soaria legítima, neste trecho.

Apesar de soarem completamente desconexos do trecho anterior, os compassos 19 a 21 possuem uma coerência interna, novamente formando um caminho com as notas através de inversões (MED, 1996).

4.4 Pop

Por fim, a última música gerada a partir de um conjunto de dados de gêneros musicais será a música Pop, cujo dataset se encontra em [datasets/pop](#). A partitura pode ser encontrada em [arquivos_resultados/partituras/pop.pdf](#), e o áudio, em [arquivos_resultados/audios/pop.wav](#).

Os compassos 19 a 24 formam progressões muito características da música Pop: dois graus (I e IV) se repetem excessivamente, mas com algumas poucas variações, deixando a música, apesar de repetitiva, leve e fácil de digerir (NUNESA J. C.; ORDANINIB; VALSESIA, 2015).

No final do compasso 26, é criada uma tensão, que é resolvida logo em seguida no compasso 27, de maneira correta. Apesar de não ser tão comum em músicas deste gênero musical, essa progressão tornou a música mais interessante harmonicamente (MED, 1996).

4.5 Frozen

Agora, analisaremos uma música gerada a partir de apenas músicas do musical *Frozen*. O dataset pode ser encontrado em [datasets/musicals/frozen](#), a partitura, em [arquivos_resultados/partituras/frozen.pdf](#), e o áudio referente à partitura pode ser encontrado em [arquivos_resultados/audios/frozen.wav](#).

Logo no início, os compassos 3 e 4 formam progressões que fazem sentido (IV7-IV-V-I) e, inclusive, formam uma cadência perfeita (V-I) (MED, 1996). No compasso seguinte, foi criada uma tensão, a partir do acorde Am7, que foi resolvido logo em seguida, formando uma progressão da forma vi7-V.

No compasso 7, chegando até o 9, o modelo reproduziu uma técnica chamada *modulação*. Isso acontece quando a música transita momentaneamente para uma tonalidade diferente na qual começou. Neste exemplo, a música modula para a tonalidade de sol maior após fazer a progressão Am7-G. A modulação é uma técnica usada muito frequentemente em musicais, geralmente no final da música, para que a peça termine num clímax. Logo em seguida, no entanto, a tonalidade de sol maior se perde e é mudada bruscamente para dó maior outra vez.

4.6 ABBA

Por fim, a última música analisada será aquela gerada pelo modelo treinado apenas com músicas do grupo musical ABBA. O dataset está em [datasets/pop/abba](#), a partitura, em [arquivos_resultados/partituras/abba.pdf](#), e o áudio pode ser encontrado em [arquivos_resultados/audios/abba.wav](#).

O primeiro aspecto que é interessante de se notar nessa música é uma clara repetição estrutural durante vários trechos. Nos compassos 5 e 6, 11 a 13, e 28 a 32, há uma certa insistência na progressão I-IV, com algumas poucas variações. Como já comentado, essa repetição estrutural é típica da música pop ([NUNESA J. C.; ORDANINIB; VALSESIA, 2015](#)).

Outra característica interessante da música gerada por este conjunto foi a presença de algumas tensões bem resolvidas. Entre os compassos 10 e 11 e dentro do compasso 17, a tensão gerada pelo acorde G/B foi resolvida de maneira correta pelo acorde de dó maior (C).

Por fim, é interessante notar que a música termina com uma cadência satisfatória (IV-I). Isso, no entanto, é uma mera coincidência já que, como já comentado, o HMM não controla esse aspecto da música.

5 Discussão

Os resultados obtidos foram, no geral, coerentes com os resultados esperados. Como previsto, as músicas geradas não possuem nenhum tipo de coerência global além daquela já garantida pela escolha dos dados, que é a tonalidade de dó maior. Isso ocorre, como já explicado, pela falta de memória do modelo utilizado. No entanto, foi possível observar a presença de diversos "blocos" de coerência local de tamanho variável.

O resultado do modelo treinado com o conjunto Global teve, de fato, o pior desempenho. Ao contrário das músicas geradas pelos outros datasets, a Global não apresentou características notáveis de nenhum gênero musical, pois todos eles estavam presentes no conjunto de dados e nenhum se destacou.

Apesar de gerarem resultados interessantes, os datasets de artistas específicos acabaram por evidenciar apenas características de seu gênero musical, não deixando explícitas características do próprio artista.

Por fim, os resultados gerados a partir dos conjuntos de dados divididos por gênero musical mostraram-se os mais bem-sucedidos por evidenciarem aspectos característicos de cada categoria. Enquanto a música de Anime apresentou acordes inesperados e progressões incomuns na música ocidental, a música Pop se manteve numa região mais homogênea, e a música de Musical, por sua vez, utilizou uma maior quantidade de inversões, que acabaram por criar até mesmo melodias dentro dos acordes.

Conclusão

Como era o esperado, os HMMs não possuem o poder necessário para gerar músicas com coerência global. Eles são capazes, no entanto, de seguir uma linha de raciocínio coerente em pequenos trechos de tamanho variável, mesmo com sua visão limitada dos passos seguidos anteriormente. Por isso, acredita-se que a capacidade de tomar decisões coerentes é o primeiro passo para a composição musical, e a adição de memória de passos anteriores aumentaria significativamente sua performance.

Uma sugestão para trabalhos futuros, portanto, seria implementar um modelo que acrescenta o aspecto de memória de ações tomadas, para que a coerência permaneça por trechos cada vez maiores, até se tornar global. A aplicação de uma LSTM, por exemplo, seria uma opção válida.

Sugere-se, também, a criação de conjuntos de dados mais amplos e mais diversos, para o teste do modelo em diferentes estilos musicais, com diferentes peculiaridades. Pode ser benéfica, também, estabelecer critérios mais rigorosos de análise harmônica para as músicas geradas, talvez, até, tornando a análise quantitativa, para que a comparação entre as peças seja menos subjetiva.

Referências

BAUM L. E.; PETRIE, T. Statistical inference for probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, v. 37, n. 6, p. 1554–1563, 2011. Citado na página 5.

DUPONT, P.; DENIS, F.; ESPOSITO, Y. Links between probabilistic automata and hidden markov models: probability distributions, learning models and induction algorithms. *Pattern Recognition*, v. 38, n. 9, p. 1349–1371, 2005. Citado na página 4.

FRANCESCHINA, J. *Music Theory Through Musical Theatre: Putting It Together*. [S.l.]: Oxford University Press, 2015. Citado na página 11.

JURKIEWICZB, M. *The Otaku Lifestyle: examining soundtrack in the Anime Canon*. Dissertação (Dissertação de Mestrado) — Faculty of the University of Missouri-Kansas City, 2019. Citado na página 11.

MED, B. *Teoria da Música*. [S.l.]: Musimed, 1996. Citado 6 vezes nas páginas 4, 6, 8, 10, 11 e 12.

NUNESA J. C.; ORDANINIB, A.; VALSESIA, F. The power of repetition: repetitive lyrics in a song increase processing fluency and drive market success. *Journal of Consumer Psychology*, v. 2, n. 25, p. 187–199, 2015. Citado 3 vezes nas páginas 7, 12 e 13.

PANZNER, M.; CIMIANO, P. Comparing hidden markov models and longshort term memory neural networks for learning action representations. 2016. Citado 2 vezes nas páginas 6 e 8.

PARDO, B.; BIRMINGHAM, W. Modeling form for on-line following of musical performances. *AAAI-05 Proc*, Júlio 2005. Citado na página 5.

RABINER, L. R. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, v. 77, n. 2, p. 257–286, 1989. Citado na página 5.

STARNER, A. P. T. *Real-Time American Sign Language Visual Recognition From Video Using Hidden Markov Models*. Dissertação (Dissertação de Mestrado) — MIT, 1995. Citado na página 5.